This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Multi-resolution dynamic mode decomposition for foreground/background separation and object tracking

J. Nathan Kutz, Xing Fu and Steven L. Brunton
Applied Mathematics
University of Washington, Seattle, WA 98195
kutz@uw.edu

N. Benjamin Erichson
Mathematics and Statistics
University of St Andrews, UK KY16 9AJ
nbe@st-andrews.ac.uk

Abstract

We demonstrate that the integration of the recently developed dynamic mode decomposition with a multi-resolution analysis allows for a decomposition of video streams into multi-time scale features and objects. A one-level separation allows for background (low-rank) and foreground (sparse) separation of the video, or robust principal component analysis. Further iteration of the method allows a video data set to be separated into objects moving at different rates against the slowly varying background, thus allowing for multiple-target tracking and detection. The algorithm is computationally efficient and can be integrated with many further innovations including compressive sensing architectures and GPU algorithms.

1. Introduction

Since the advent of scientific computing, matrix decomposition techniques have dominated many of the transformative algorithms used in applications across the engineering, biological and physical sciences. Indeed, efficient computation of large scale systems almost always depends upon taking advantage of a matrix decomposition in order to either leverage low-rank structure, sparsity or an efficient representation, for instance. In the specific application of video analysis, the time snapshots of video streams are used to compose matrices that are high-dimensional, but which often have a high-degree of correlation between frames. Understanding the correlation structure between time frames is fundamental for accurate and real-time video surveillance techniques. For instance, removing *background* variations in a video stream, which typically are highly correlated between frames, are at the forefront of modern data-analysis research. Background/foreground separation is typically an integral step in detecting, identifying, tracking, and recognizing objects in streaming video streams. We show that a recent innovation from dynamical systems theory, the *Dynamic Mode Decomposition (DMD)* [23, 24, 9, 22, 30, 16], provides a decomposition of data into spatio-temporal modes that correlates the data across spatial features (like principal component analysis (PCA)), but also pins the correlated data to unique temporal Fourier modes. This method easily distinguishes the stationary background from the dynamic foreground by differentiating between the near-zero temporal Fourier modes and the remaining modes bounded away from the origin, respectively. We demonstrate that the method can be generalized for tracking objects, thus providing a principled approach to video diagnostics and target detection.

For computer vision applications integrating video feeds, algorithms are envisioned to be implemented in real-time on high-definition video streams. The algorithms must not only be extremely fast to handle the data demand, but also must be robust enough to handle diverse, complicated, and cluttered backgrounds. Methods often need to be flexible enough to adapt to changes in a scene due to, for instance, illumination changes that can occur naturally throughout the day, or potential location changes for portable devices. Given the importance of this task for surveillance and target tracking/aquisition, a variety of matrix decomposition techniques have already been developed. For instance, a number of iterative (optimization and gradient descent based) techniques have already been developed in order to perform background/foreground separation [18, 28, 19, 13, 8]. We point the reader to several recent reviews [1, 2, 25, 26, 3] and a textbook [27] which highlight many of the methods developed and their performance metrics.

As a matrix separation problem, the task is to separate the video data into *low-rank* (background) and *sparse* (foreground) components. The importance of this viewpoint was realized by Candès et al. in the framework of *robust principal component analysis* (RPCA) [8]. By weighting a combination of the nuclear and the L^1 norms, a convenient convex optimization problem (*principal component pursuit*) was demonstrated, under suitable assumptions, to recover the low-rank and sparse components ex-

actly of a given data-matrix (or video for our purposes). It was also compared to the state-of-the-art computer vision procedure developed by De La Torre and Black [15]. We advocate a similar matrix separation approach, but by using DMD [23, 24, 9, 22, 30, 16]. Since the method ties the spatial correlation of pixels to temporal Fourier dynamics, the zero mode represents the stationary, or low-rank, background. Although it was originally introduced in the fluid mechanics community, DMD has emerged as a powerful tool for analyzing the dynamics of nonlinear systems [23, 24, 9, 22, 30, 16], including those in neuroscience [5] and financial trading [20].

More broadly, video streams often are often comprised of multi-scale temporal and/or spatial features of interest. This is also true in many multi-scale systems that pervade the engineering, biological and physical sciences. The DMD method can be used as a transformative tool of innovation in such problems since it can circumvent the significant challenges in efficiently connecting micro- to macro-scale effects that are separated potentially by orders of magnitude spatially and/or temporally [17]. Wavelet-based methods and/or windowed Fourier Transforms are ideally structured to perform such multi-resolution analyses (MRA) as they systematically remove temporal or spatial features by a process of recursive refinement of sampling from the data of interest. Typically, MRA is performed on either space or time, but not both simultaneously. By integrating the concept of MRA with the DMD, a Multi-Resolution DMD (MRDMD) is developed and shown to naturally integrate space and time so that the multi-scale spatio-temporal features are easily separated. This allows for a separation of objects of interest in video feeds that are evolving temporally at different rates. For instance, in a video feed with a person walking and a car driving by, it is envisioned that three separate feeds would be created: the background (no temporal evolution), a video of the walker (slow temporal evolution) and a car (fast temporal evolution). The MRDMD allows for this decomposition and analysis of video feeds in a real-time architecture.

2. Dynamical Systems and Decompositions

The DMD method emerged from the dynamical systems literature, with specific applications in modeling complex fluid flows. In this context, it is assumed that there is some driving dynamical system generating the observed data. For video feeds, we don't expect this to be true. For instance, in the example video of a person walking and a car driving by, dynamics are not prescribed by some set of governing equations. However, DMD reconstructs the best linear dynamical system modeling these features.

One may consider the DMD as a way to approximate the dynamics of a nonlinear system:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t)\,. \tag{1}$$

In addition, both measurements of the system $g(\mathbf{x}, t) = 0$, and initial conditions are prescribed $\mathbf{x}(0) = \mathbf{x}_0$. Typically $\mathbf{x}$ is an N-dimensional vector ($N \gg 1$) that arises from either discretization of a complex system, or in the case of video streams, it is the total number of pixels in a given frame. The governing equation and initial condition specify a well-posed initial value problem. The inclusion of measurements $g(\mathbf{x}, t)$, let's say M of them, make the system overdetermined. By including model error along with noisy measurements, one can formulate an optimal predictive strategy using data-assimilation and Kalman filtering innovations.

In general the solution of the governing nonlinear evolution is not possible to construct since it is unknown, especially for video applications. In the DMD framework, the snapshot measurements and initial conditions alone are used to approximate the dynamics and predict the future state. The DMD procedure thus constructs the proxy, approximate linear evolution

$$\frac{d\tilde{\mathbf{x}}}{dt} = \mathbf{A}\tilde{\mathbf{x}} \tag{2}$$

with $\tilde{\mathbf{x}}(0) = \tilde{\mathbf{x}}_0$ and whose solution is

$$\tilde{\mathbf{x}}(t) = \sum_{k=1}^{K} b_k \psi_k \exp(\omega_k t) \tag{3}$$

where ψ_k and ω_k are the eigenfunctions and eigenvalues of the matrix $\mathbf{A}$. The ultimate goal in the DMD algorithm is to optimally construct the matrix $\mathbf{A}$ so that the true and approximate solution remain optimally close for true solution in a least-square sense:

$$\|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)\| \ll 1. \tag{4}$$

Of course, the optimality of the approximation holds only over the sampling window where $\mathbf{A}$ is constructed, but the approximate solution can be used to not only make future state predictions, but also decompose the dynamics into various time-scales since the ω_k are prescribed and have true temporal meaning. Moreover, the DMD makes use of low-rank structure so that the total number of modes, $K \ll N$, allows for dimensionality reduction of the video stream.

At its core, the DMD method can be thought of as an ideal combination of spatial dimensionality-reduction techniques, such as PCA, with Fourier Transforms in time. Interpreting a video stream in this context allows for background/foreground separation. It also allows for further innovations that integrate the DMD with key concepts from wavelet theory and MRA. Specifically, the DMD method

takes snapshots of video streams, with sampling windows of variable frequency and duration, in order to leverage ideas from wavelet theory that sifts out information at different scales. Indeed, an iterative refinement of progressively shorter snapshot sampling windows and recursive extraction of DMD modes from slow- to increasingly-fast time scales allows for a MRDMD that allows for object tracking of video features evolving on different timescales.

3. Dynamic Mode Decomposition

The DMD method provides a spatio-temporal decomposition of data into a set of dynamic modes that are derived from snapshots or measurements of a given system in time. The mathematics underlying the extraction of dynamic information from time-resolved snapshots is closely related to the idea of the Arnoldi algorithm [23], one of the workhorses of fast computational solvers. The data collection process involves two parameters:

N = number of spatial points saved per time snapshot
M = number of snapshots taken

Originally the algorithm was designed to collect data at regularly spaced intervals of time. However, new innovations allow for both sparse spatial [6] and temporal collection of data as well as irregularly spaced collection times. Indeed, Tu et al. [30] gives the best definition of the DMD:

Definition: Dynamic Mode Decomposition (Tu et al. 2014 [30]): *Suppose we have a dynamical system (1) and two sets of data*

$$\mathbf{X}=\begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_M \\ | & | & & | \end{bmatrix}, \quad \mathbf{X}'=\begin{bmatrix} | & | & & | \\ \mathbf{x}'_1 & \mathbf{x}'_2 & \cdots & \mathbf{x}'_M \\ | & | & & | \end{bmatrix} \tag{6}$$

with $\mathbf{x}_k$ an initial condition to (1) and $\mathbf{x}'_k$ it corresponding output after some prescribed evolution time τ with there being m initial conditions considered. The DMD modes are eigenvectors of

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger} \tag{7}$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse.

The DMD method approximates the modes of the so-called *Koopman operator*. The Koopman operator is a linear, infinite-dimensional operator that represents nonlinear, infinite-dimensional dynamics without linearization [22, 21], and is the adjoint of the Perron-Frobenius operator. The method can be viewed as computing, from the experimental data, the eigenvalues and eigenvectors (low-dimensional modes) of a linear model that approximates the underlying dynamics, even if the dynamics is nonlinear. Since the model is assumed to be linear, the decomposition gives the growth rates and frequencies associated with each mode. If the underlying model is linear, then the DMD method recovers the leading eigenvalues and eigenvectors computed using solution methods for linear differential equations.

Mathematically, the Koopman operator $\mathbf{A}$ is a linear, time-independent operator $\mathbf{A}$ such that

$$\mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j \tag{8}$$

where j indicates the specific data collection time and $\mathbf{A}$ is the linear operator that maps the data from time t_j to t_{j+1}. The vector $\mathbf{x}_j$ is an N-dimensional vector of the data points collected at time j. The computation of the Koopman operator is at the heart of the DMD methodology. It should be noted that this is different than linearizing the dynamics.

In practice, when the state dimension N is large, the matrix $\mathbf{A}$ may be intractable to analyze directly. Instead, DMD circumvents the eigendecomposition of $\mathbf{A}$ by considering a rank-reduced representation in terms of a projected matrix $\tilde{\mathbf{A}}$. The DMD algorithm proceeds as follows [30]:

1. Decompose the data matrix $\mathbf{X}$ via an SVD [29]:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{9}$$

where $*$ denotes the conjugate transpose, $\mathbf{U} \in \mathbb{C}^{N\times K}$, $\mathbf{\Sigma} \in \mathbb{C}^{K\times K}$ and $\mathbf{V} \in \mathbb{C}^{M-1\times K}$. Here K is the rank of the reduced SVD approximation to $\mathbf{X}$. The left singular vectors $\mathbf{U}$ are like PCA modes.

The SVD reduction in (9) could also be used for a low-rank truncation of the data using, for instance, a principled way to truncate noisy data [11].

2. Compute $\tilde{\mathbf{A}}$, the $K \times K$ projection of the full matrix $\mathbf{A}$ onto low-rank modes of $\mathbf{U}$:

$$\begin{aligned} \mathbf{A} &= \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^* \\ \implies \tilde{\mathbf{A}} &= \mathbf{U}^*\mathbf{A}\mathbf{U} = \mathbf{U}^*\mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}. \end{aligned} \tag{10}$$

3. Eigendecompose $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}, \tag{11}$$

where columns of $\mathbf{W}$ are eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix containing the corresponding eigenvalues λ_k.

4. Reconstruct the eigendecomposition of $\mathbf{A}$ from $\mathbf{W}$ and $\mathbf{\Lambda}$. In particular, the eigenvalues of $\mathbf{A}$ are given by $\mathbf{\Lambda}$ and the eigenvectors of $\mathbf{A}$ (DMD modes) are given by columns of $\mathbf{\Psi}$:

$$\mathbf{\Psi} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}. \tag{12}$$

Note that Eq. (12) from [30] differs from the formula $\mathbf{\Psi} = \mathbf{UW}$ from [23], although these will tend to converge if $\mathbf{X}$ and $\mathbf{X}'$ have the same column spaces.

With the low-rank approximations of both the eigenvalues and eigenvectors in hand, the projected future solution can be constructed for all time in the future. By first rewriting for convenience $\omega_k = \ln(\lambda_k)/\Delta t$, where Δt is the time between frames, then the approximate solution at all future times, $\tilde{\mathbf{x}}(t)$, is given by

$$\tilde{\mathbf{x}}(t) = \sum_{k=1}^{K} b_k(0)\psi_k(\boldsymbol{\xi})\exp(\omega_k t) = \mathbf{\Psi}\text{diag}(\exp(\boldsymbol{\omega} t))\mathbf{b} \quad (13)$$

where $\boldsymbol{\xi}$ are the spatial coordinates, $b_k(0)$ is the initial amplitude of each mode, $\mathbf{\Psi}$ is the matrix whose columns are the eigenvectors ψ_k, $\text{diag}(\omega t)$ is a diagonal matrix whose entries are the eigenvalues $\exp(\omega_k t)$, and $\mathbf{b}$ is a vector of the coefficients b_k.

An alternative interpretation of (13): it is the least-square fit, or regression, of a linear dynamical system $d\tilde{\mathbf{x}}/dt = \mathbf{A}\tilde{\mathbf{x}}$ to the data sampled much as suggested in (4). For a multi-resolution analysis, each level of the multi-scale decomposition produces a linear dynamical system, or matrix $\mathbf{A}$, for the time-scale under consideration.

It only remains to compute the initial coefficient values $b_k(0)$. If we consider the initial snapshot ($\mathbf{x}_1$) at time $t_1 = 0$, let's say, then (13) gives $\mathbf{x}_1 = \mathbf{\Psi}\mathbf{b}$. This generically is not a square matrix so that its solution

$$\mathbf{b} = \mathbf{\Psi}^{\dagger}\mathbf{x}_1 \quad (14)$$

can be found using a pseudo-inverse. Indeed, $\mathbf{\Psi}^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. The pseudo-inverse is equivalent to finding the best solution $\mathbf{b}$ the in the least-squares (best fit) sense. This is equivalent to how DMD modes were derived originally.

4. Robust PCA with DMD

For a given data matrix, perhaps generated from a nonlinear dynamical system such as (1), the RPCA method will seek out the sparse structures within the data, while simultaneously fitting the remaining entries to a low-rank (highly correlated) basis. As long as the given data is truly of this nature, i.e., it is a superposition of a component that lies in a low-dimensional subspace and a sparse component, then the RPCA algorithm has been proven by Candès et al. [8] to perfectly separate the given data $\mathbf{X}$ such that

$$\mathbf{X} = \mathbf{L} + \mathbf{S}\,, \quad (15)$$

where $\mathbf{L}$ is low-rank and $\mathbf{S}$ is sparse. The key to the RPCA algorithm is formulating this specific problem into a tractable, nonsmooth convex optimization problem known as *principal component pursuit* (PCP) [3].

For DMD, the separation relies on the interpretation of the ω_k frequencies in the DMD solution reconstructions represented in general by (3), and more specifically as in (13). In particular, low-rank features in video, for instance, are such that $|\omega_j| \approx 0$, i.e. they are slowly changing in time. Thus if one sets a threshold ϵ so as to gather all the slow, low-rank modes where $|\omega_j| \leq \epsilon$, then the separation can be accomplished. The selection of the threshold value ϵ is chosen to select out the stationary (zero mode) and potential quasi-stationary (near zero mode(s)) behavior of the video stream. The total number of snapshots collected would guide the selection of the threshold value. This reproduces a representation of the $\mathbf{L}$ and $\mathbf{S}$ matrices of the form:

$$\mathbf{L} \approx \sum_{|\omega_k|\leq\epsilon} b_k\psi_k\exp(\omega_k t)\,,\ \ \mathbf{S} \approx \sum_{|\omega_k|>\epsilon} b_k\psi_k\exp(\omega_k t)\,. \quad (16)$$

Note that the low-rank matrix $\mathbf{L}$ picks out only a small number of the total number of DMD modes to represent the *slow* oscillations or DC content in the data ($\omega_j = 0$). The DC content is exactly the background mode when interpreted in the video stream context with a fixed and stable camera. The advantage of the DMD method and its sparse/low-rank separation is the computational efficiency of achieving (16), especially when compared to the optimization methods of RPCA, i.e. a single SVD versus an SVD at each iteration step. A demonstration of the performance is given in Sec. 6.

5. Multi-Resolution Analysis of Video

The MRDMD recursively removes low-frequency, or slowly-varying, content from a given collection of snapshots, making it ideal for separating different time-scale features in video. Typically, the number of snapshots M are chosen so that the DMD modes provide an approximately full rank approximation of the dynamics observed. Thus M is chosen so that all high- and low-frequency content is present. In the MRDMD, M is originally chosen in the same way so that an approximate full rank approximation can be accomplished. However, from this initial pass through the data, the slowest m_1 modes are removed, and the domain is divided into two segments with $M/2$ snapshots each. DMD is once again performed on each $M/2$ snapshot sequences. Again the slowest m_2 modes are removed and the algorithm is continued until a desired termination.

MRDMD approximates the solution (13) as:

$$\mathbf{x}_{\text{mrDMD}}(t) = \sum_{k=1}^{M} b_k(0)\psi_k^{(1)}(\boldsymbol{\xi})\exp(\omega_k t) \quad (17)$$

$$= \underbrace{\sum_{k=1}^{m_1} b_k(0)\psi_k^{(1)}(\boldsymbol{\xi})\exp(\omega_k t)}_{\text{(slow modes)}} + \underbrace{\sum_{k=m_1+1}^{M} b_k(0)\psi_k^{(1)}(\boldsymbol{\xi})\exp(\omega_k t)}_{\text{(fast modes)}}$$

where the $\psi_k^{(1)}(\mathbf{x})$ represent the DMD modes computed from the full M snapshots.

The first sum in this expression is the slow-mode dynamics whereas the second sum is the faster time-scale dynamics. The second sum can be computed to yield the fast scale data matrix:

$$\mathbf{X}_{M/2} = \sum_{k=m_1+1}^{M} b_k(0)\psi_k^{(1)}(\boldsymbol{\xi})\exp(\omega_k t)\,. \tag{18}$$

The DMD analysis outlined in the previous section can now be performed once again on the data matrix $\mathbf{X}_{M/2}$. However, the matrix $\mathbf{X}_{M/2}$ is now separated into two matrices

$$\mathbf{X}_{M/2} = \mathbf{X}_{M/2}^{(1)} + \mathbf{X}_{M/2}^{(2)} \tag{19}$$

where the first matrix contains the first $M/2$ snapshots and the second matrix contains the remaining $M/2$ snapshots. The m_2 slow-DMD modes at this level are given by $\psi_k^{(2)}$, where they are computed separately in the first of second interval of snapshots.

The iteration process works by recursively removing slow frequency components and building the new matrices $\mathbf{X}_{M/2}, \mathbf{X}_{M/4}, \mathbf{X}_{M/8}, \cdots$ until a desired multi-resolution decomposition has been achieved. The approximate MRDMD solution can then be constructed as follows:

$$\begin{aligned}\mathbf{x}_{\text{mrDMD}}(t) = &\sum_{k=1}^{m_1} b_k^{(1)}\psi_k^{(1)}\exp(\omega_k^{(1)}t) \\ &+\sum_{k=1}^{m_2} b_k^{(2)}\psi_k^{(2)}\exp(\omega_k^{(2)}t) + \sum_{k=1}^{m_3} b_k^{(3)}\psi_k^{(3)}\exp(\omega_k^{(3)}t) + \cdots\end{aligned} \tag{20}$$

where at the evaluation time t, the correct modes from the sampling window are selected at each level of the decomposition. Specifically, the $\psi_k^{(k)}$ and $\omega_k^{(k)}$ are the DMD modes and DMD eigenvalues at the kth level of decomposition, the $b_k^{(k)}$ are the initial projections of the data onto the time interval of interest, and the m_k are the number of slow-modes retained at each level. The advantage of this method is readily apparent: different spatio-temporal DMD modes are used to represent key multi-resolution features. Thus there is not a single set of modes that dominates the SVD and potentially marginalizes features at other time scales.

Figure 1 illustrates the multi-resolution DMD process pictorially. In the figure, a three-level decomposition is performed with the slowest scale represented in blue (eigenvalues and snapshots), the mid-scale in red and the fast scale in green. Such an example may correspond to the example video stream attempting to extract the background and two objects moving at different speeds, a pedestrian and a car, for instance. The connection to multi-resolution wavelet analysis is also evident from the bottom panels as one can see that the mrDMD method successively pulls out time-frequency information in a principled way. The sampling strategy can be easily modified so as to sample a fixed number, for instance M, data snapshots in each sampling window. The value of M need not be large as only the slow modes need to be resolved. Thus the sampling rate (in real time units) would increase as the decomposition proceeds from one level to the next.

5.1. Formal mrDMD Expansion

To construct the MRDMD solution, one must account for the number of levels (L) of the decomposition, the number of time bins (J) for each level, and the number of modes retained at each level (m_L):

$$\begin{aligned}&\ell = 1, 2, \cdots, L \;\text{ number of decomposition levels}\\ &j = 1, 2, \cdots, J \;\text{ number time bins per level } (J = 2^{(\ell-1)})\\ &k = 1, 2, \cdots, m_L \;\text{ number of modes extracted at level } L.\end{aligned}$$

To formally define the series solution for $\mathbf{x}_{\text{mrDMD}}(t)$, the indicator function is used

$$f_{\ell,j}(t) = \begin{cases} 1 & t \in [t_j, t_{j+1}] \\ 0 & \text{elsewhere}\end{cases} \quad \text{with} \quad j = 1, 2, \cdots, J \tag{22}$$

where $J = 2^{(\ell-1)}$. This is only non-zero in the interval, or time bin, associated with the value of j. The parameter ℓ denotes the level of the decomposition.

The three indices and indicator function (22) give the MRDMD solution expansion

$$\mathbf{x}_{\text{mrDMD}}(t) = \sum_{\ell=1}^{L}\sum_{j=1}^{J}\sum_{k=1}^{m_L} f_{\ell,j}(t) b_k^{(\ell,j)}\psi_k^{(\ell,j)}(\boldsymbol{\xi})\exp(\omega_k^{(\ell,j)}t)\,. \tag{23}$$

This is a concise definition of the MRDMD solution that includes the information on the level, time bin location and number of modes extracted. Figure 2 demonstrates the mrDMD decomposition in terms of the solution (23). In particular, each mode is represented in its respective time bin and level. An alternative interpretation of this solution is that it yields the least-square fit, at each level ℓ of the decomposition, to the linear dynamical system

$$\frac{d\mathbf{x}^{(\ell,j)}}{dt} = \mathbf{A}^{(\ell,j)}\mathbf{x}^{(\ell,j)} \tag{24}$$

where the matrix $\mathbf{A}^{(\ell,j)}$ captures the dynamics in a given time bin j at level ℓ.

In connecting this to sparse and low-rank decompositions, the MRDMD is equivalent to producing a series of decompositions at each level of resolution where

$$\mathbf{X}^{(\ell,j)} = \mathbf{L}^{(\ell,j)} + \mathbf{S}^{(\ell,j)}\,. \tag{25}$$

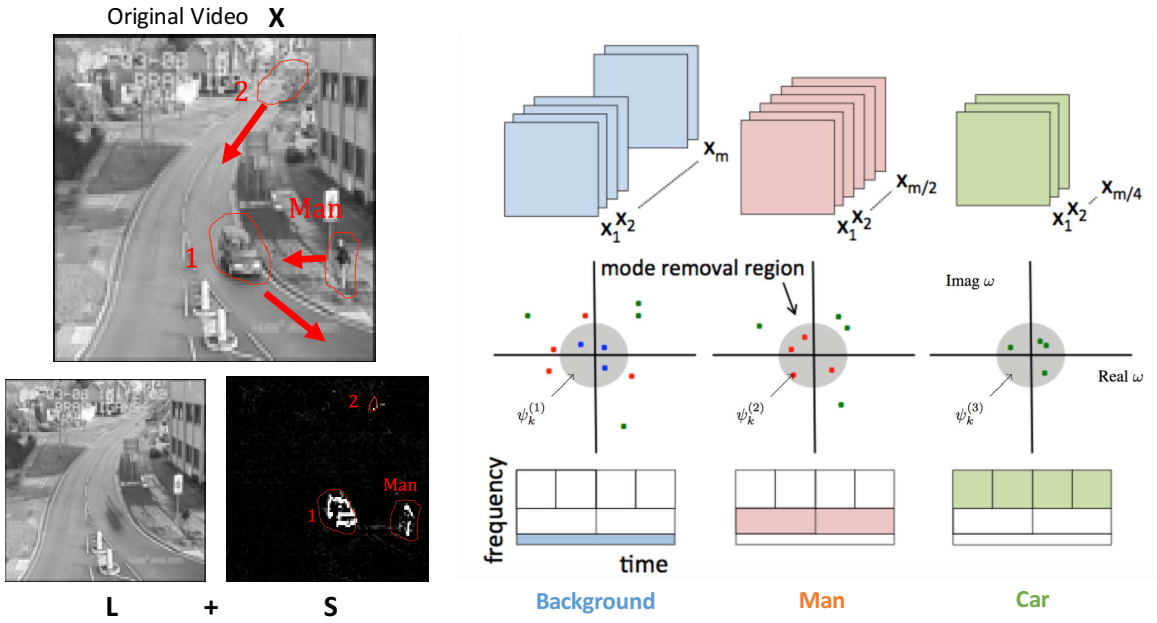

Figure 1: Representation of the multi-resolution dynamic mode decomposition on an example video (top left) that includes three different time scale features (annotated), a background, a pedestrian (slow) and a car (fast). A standard DMD foreground/background separation of the video $\mathbf{X} = \mathbf{L} + \mathbf{S}$ is shown in the bottom left panels. The MRDMD with successive sampling of the data, initially with M snapshots and decreasing by a factor of two at each resolution level, is shown on the right. The DMD spectrum is shown in the middle panel right where there are m_1 (blue dots) slow-dynamic modes (background) at the slowest level, m_2 (red) modes at the next level (man) and m_3 (green) modes at the fastest (car) time-scale shown. The shaded region represents the modes that are removed at that level. The bottom right panels shows the wavelet-like time-frequency decomposition of the data color coded with the snapshots and DMD spectral representations.

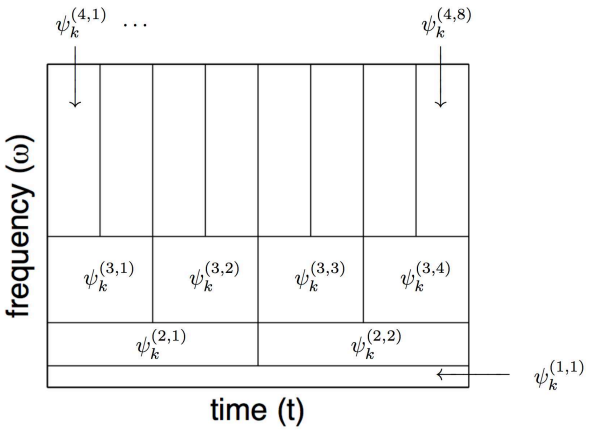

Figure 2: The MRDMD mode decomposition and hierarchy. Represented are the modes $\psi_k^{\ell,j}(\boldsymbol{\xi})$ and their position in the decomposition structure. The integer values, ℓ, j and k, uniquely express the time level, bin and decomposition.

Thus the decomposition is recursive in nature. In this formulation, we can alternatively rewrite (23) using (16) as

$$\mathbf{x}_{\text{mrDMD}}(t) = \mathbf{L}^{(1,j)} + \mathbf{L}^{(2,j)} + \mathbf{L}^{(3,j)} + \mathbf{S}^{(3,j)} \tag{26}$$

where a 3-level truncation is assumed.

The indicator function $f_{\ell,j}(t)$ acts as sifting function for each time bin. Interestingly, this function acts as the Gabór window of a windowed Fourier transform [16]. Since our sampling bin has a hard cut-off of the time series, it may introduce some artificial high-frequency oscillations. Time-series analysis, and wavelets in particular, introduce various functional forms that can be used in an advantageous way. Thus thinking more broadly, one can imagine using wavelet functions for the sifting operation, thus allowing the time function $f_{\ell,j}(t)$ to take the form of one of the many potential wavelet basis, i.e. Haar, Daubechies, Mexican Hat, etc. This will be considered in future work. For the present, we simply use the sifting function introduced in (22)

5.2. Object Tracking

To demonstrate the efficacy of the MRDMD method, we first construct a video example that is comprised of four different video feeds: a background and three different tempo-

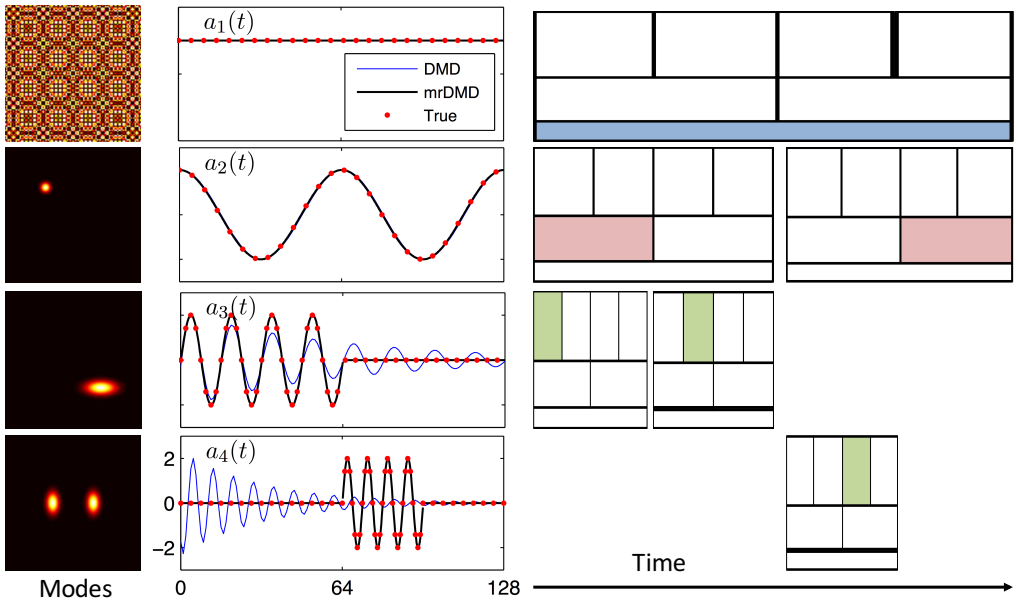

Figure 3: Demonstrate of the MRDMD for decomposing a stream of four unique video streams (left panels) that are constructed to have different time-scale dynamics (middle panels). Also included in the middle panel are the reconstructed time dynamics from the MRDMD method versus the standard DMD technique. The MRDMD does an exceptional job of separating the time scales and capturing transient dynamics. The right panels illustrate the regions in time-frequency space where relevant information for each mode is found, giving a visual diagnostic for how information is encoded in the video stream.

ral objects (features). Figure 3 shows the specific modes combined. Such a video helps build intuition about how the decomposition works, especially in relation to the standard DMD method. Thus we combine the four modes shown in the left panels with the time dynamics given in the middle panels. The four modes used to construct the true solution are represented by $\bar{\psi}_j$ for $j = 1, 2, 3$ and 4. Their corresponding time dynamics are given by $a_j(t)$. Thus the true solution is expressed by

$$\mathbf{X} = \sum_{j=1}^{4} a_j(t)\psi_j \,. \tag{27}$$

DMD (represented by $\mathbf{x}_{\text{DMD}}$ and the modes ψ_j of (13) with $j = 1, 2, 3$ and 4) and mrDMD (represented by $\mathbf{x}_{\text{mrDMD}}$ and the $\psi_k^{(\ell,j)}$ of (20) where $k=1$ and $\ell=1, 2, 3$) reconstruct $\bar{\mathbf{x}}$.

The success of the MRDMD algorithm suggests how one can track objects that have different temporal signatures. For instance, Fig. 1 already shows that a realistic video with both cars and pedestrians is ideally suited for the method as it can distinguish between slow moving pedestrians and rapidly moving cars. The example of Fig. 3 shows that a good separation can be achieved in such a situation provided the objects of interest have a time-scale separation.

6. Experimental Evaluation

The DMD is evaluated on seven gray scaled videos belonging to four different categories (each representing a specific challenge) of the ChangeDetection.net benchmark dataset [31]. The binary foreground (classification) mask can be obtained by thresholding the euclidean distance between the modeled background and the original video frame. The performance is then quantified by different statistical metrics like F-measure, recall, precision, percentage of wrong classifications (PWC), false positive rate (FPR), false negative rate (FNR) and specificity [31, 7]. Table 1 summarizes the results and Figure 4 shows a sample frame.

In our experimental setting, the target rank $k = 15$ was used for all videos. Additionally, a soft-threshold based on Lasso was used to select the low-rank modes, instead of defining a hard threshold ϵ in advance [14]. Finally, for smoothing the foreground mask and reducing noise a median filter was applied. The archived results are highly competitive and show the ability of DMD for background/foreground separation. A primary advantage is the computational time (e.g. ~ 55 fps for a 320x240 video) of DMD in comparison with most other state-of-the-art algorithms that are capable of computing 10 to 25 frames per second (fps). However, the performance depends on the length of the used video sequence, hence the number of frames is a trade-off between speed and accuracy. In fu-

Measure	Baseline		Bad Weather		Dynamic Background	Intermit. Obj. Motion	
	Highway	Pedestrian	Skating	Blizzard	Canoe	Sofa	Parking
F-Measure	0.918	0.960	0.917	0.864	0.921	0.674	0.895
Recall	0.911	0.975	0.853	0.843	0.895	0.617	0.878
Precision	0.924	0.946	0.993	0.886	0.950	0.744	0.913
PWC	0.971	0.079	0.760	0.310	0.540	2.601	1.586
FPR	0.005	0.001	0.000	0.001	0.002	0.010	0.007
FNR	0.089	0.025	0.147	0.157	0.105	0.383	0.122
Specifity	0.995	0.999	0.999	0.999	0.998	0.990	0.993
Evaluated frames	1230	800	3100	5000	389	2250	1400

Table 1: Evaluation results.

(a) Original frame (b) Ground truth (c) Modeled background (d) Foreground mask

Figure 4: Illustration of background/foreground separation using DMD.

ture research we will investigate the performance on a more comprehensive set of videos in order to evaluate DMD with heavily crowded scenes or camera jitter, for instance.

7. Conclusions and Outlook

By interpreting video streams as a dynamical system, the power of the DMD algorithm can be leveraged to decompose video data into a set of dynamic modes that are derived from individual snap shots. It also leverages ideas from wavelet theory and multi-resolution analysis, allowing for a principled reconstruction of multi-resolution, spatio-temporal video feeds. The effectiveness of the method is demonstrated on several example data sets, highlighting its ability to extract critical information and enact data-driven discovery protocols for background removal and multiple target detection. The method can be viewed as computing, from the snapshots alone, the eigenvalues and eigenvectors (low-dimensional modes) of a linear model that approximates the underlying video dynamics. By interpreting the DMD eigenvalues as corresponding to prescribed time scale dynamics, one can extract spatio-temporal structures recursively for shorter and shorter sampling windows. Thus the slow-modes are removed first and the data is filtered for analysis of its higher frequency content. This recursive sampling structure is demonstrated to be effective in allowing for a reconstruction of different time-scale features.

One of the most attractive features of the DMD algorithm is the number of enhancements and innovations around the basic decomposition scheme. This can help performance in three specific areas: (i) accuracy of foreground/background subtraction as well as object detection, (ii) computational speed and (iii) memory requirements. In what follows, a list of innovations are highlighted that are capable of greatly enhancing the MRDMD algorithm. **Compressive Sampling:** Natural images are known to be sparse in many basis functions, e.g. wavelets. The DMD architecture can easily capitalize on this fact by sub-sampling of the pixel space. For DMD, this has been recently demonstrated to work well for understanding dynamical systems [6]. This helps the MRDMD with both speed and memory requirements. **Incremental and Random SVDs:** Background and foreground objects do not typically change significantly from frame to frame, thus allowing for incremental updates (incremental SVD [4]) of the background and foreground objects. Further, random SVD architectures [12] can also be utilized to improve computational efficiency. **Denoising:** Recent innovations in DMD theory suggest that a principled and effective approach can be taken to removing noise from the data matrix $\mathbf{X}$ before performing a DMD decomposition [10]. Such a step is critical in evaluating video streams, especially if noise removal happens in a recursive, multi-time scale manner. **GPU Architectures:** The DMD algorithm is also amenable to efficient implementation on GPU platforms, allowing for real-time target tracking and foreground/background separation on HD video feeds. One can easily envision applications where the MRDMD architecture is integrated in hardware, and the GPU provides an effective strategy for providing real-time analysis of objects.

References

[1] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. *Comparative Study of Background Subtraction Algorithms. Journal of Electronic Imaging 19 (2010)*, 19(3):033003, 2010.

[2] T. Bouwman. *Recent advanced statistical background modeling for foreground detection: a systematic survey. RPCS*, 4(3):147–176, 2011.

[3] T. Bouwman and E. H. Zahzah. *Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance. Comp. Vis. Imag. Under.*, 122:22–34, 2014.

[4] M. Brand. *Fast Low-Rank Modifications of the Think Singular Value Decomposition . MITSUBISHI ELECTRIC RESEARCH LABORATORIES*, TR2006-059, 2006.

[5] B. Brunton, L. Johnson, J. Ojemann, and J. N. Kutz. *Extracting Spatial-Temporal Coherent Patterns in Large-Scale Neural Recordings Using Dynamic Mode Decomposition. arXiv:1409.5496*, 2015.

[6] S. Brunton, J. Proctor, and J. N. Kutz. *Compressive sampling and dynamic mode decomposition. Journal of Computational Dynamics*, to appear (2015).

[7] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1937–1944. IEEE, 2011.

[8] E. Candès, X. Li, Y. Ma, and J. Wright. *Robust Principal Component Analysis? Computing Research Repository*, abs/0912.3599, 2009.

[9] K. Chen, J. Tu, and C. Rowley. *Variants of Dynamic Mode Decomposition: Boundary Condition, Koopman, and Fourier Analyses. Journal of Nonlinear Science*, 22(6):887–915, 2012.

[10] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. *Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. arXiv:1507.02264*, 2015.

[11] M. Gavish and D. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *Information Theory, IEEE Transactions on*, 60(8):5040–5053, Aug 2014.

[12] N. Halko, P. G. Martinsson, and J. A. Tropp. *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev.*, 53(2):217–288, 2011.

[13] J. He, L. Balzano, and A. Szlam. *Incremental Gradient on the Grassmannian for Online Foreground and Background Separation in Subsampled Video.* In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1568–1575, 2012.

[14] M. Jovanovic, P. Schmid, and J. Nichols. Low-rank and sparse dynamic mode decomposition. *Center for Turbulence Research Annual Research Briefs*, pages 139–152, 2012.

[15] M. Kleinsteuber, F. Seidel, and C. Hage. *pROST: A Smoothed ℓ_p-Norm Robust Online Subspace Tracking Method for Realtime Background Subtraction in Video. Machine Vision and Applications, Special Issue on Background Modeling for Foreground Detection in Real-World Dynamic Scenes*, 2013.

[16] J. N. Kutz. *Data-driven modeling and scientific computing: Methods for Integrating Dynamics of Complex Systems and Big Data.* Oxford Press, 2013.

[17] J. N. Kutz, X. Fu, and S. Brunton. *Multi-Resolution Dynamic Mode Decomposition. arXiv:1506.00564*, 2015.

[18] L. Li, W. Huang, I. Gu, and Q. Tian. *Statistical Modeling of Complex Backgrounds for Foreground Object Detection. IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[19] L. Maddalena and A. Petrosino. *A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[20] J. Mann and J. N. Kutz. *Dynamic Mode Decomposition for Financial Trading Strategies. arXiv:1508.04487*, 2015.

[21] I. Mezić. *Analysis of Fluid Flows via Spectral Properties of the Koopman Operator. Annual Review of Fluid Mechanics*, 45:357–378, 2013.

[22] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. *Spectral analysis of nonlinear flows. Journal of Fluid Mechanics*, 641:115–127, 2009.

[23] P. Schmid. *Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics*, 656:5–28, 2010.

[24] P. Schmid, L. Li, M. Juniper, and O. Pust. *Applications of the dynamic mode decomposition. Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[25] M. Shah, J. Deng, and B. Woodford. *Video Background Modeling: Recent Approaches, Issues and Our Solutions. Machine Vision and Applications, Special Issue on Background Modeling fro Foreground Detection in Real-World Dynamics*, 25:1105–1119, 2014.

[26] A. Shimada, Y. Nonaka, H. Nagahara, and R. Taniguchi. *Case Based Background Modeling-Towards Low-Cost and High-Performance Background Model. Machine Vision and Applications, Special Issue on Background Modeling fro Foreground Detection in Real-World Dynamics*, 25:1121–1131, 2015.

[27] E. T. Bouwman. *Handbook on Robust Decomposition in Low Rank and Sparse Matrices and its Applications in Image and Video Processing.* CRC Press, 2015.

[28] Y. Tian, M. Lu, and A. Hampapur. *Robust and Efficient Foreground Analysis for Real-Time Video Surveillance.* In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, volume 1, pages 1182–1187, 2005.

[29] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, Philadelphia, 1997.

[30] J. Tu, C. Rowley, D. Luchtenberg, S. Brunton, and J. N. Kutz. *On Dynamic Mode Decomposition: Theory and Applications. Journal of Computational Dynamics*, 1:391–421, 2014.

[31] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. Cdnet 2014: an expanded change detection benchmark dataset. In *IEEE Workshop on Computer Vision and Pattern Recognition*, pages 393–400. IEEE, 2014.